Paradigm	System	Code increase	Memory increase	Parallel Performance					
MPI Message-passing	T3E	100%	70%	P	Refine		Overhead		Total
				8	4.53		14.44		18.97
				160	0.61		2.39		**3.00**
				512	0.14		4.95		5.09
	Origin 2000	100%	70%	P	Refine		Overhead		Total
				8	8.31		11.62		19.93
				64	1.41		3.99		**5.40**
Shared-memory directives	Origin 2000	10%	5%	P	Refine		Overhead		Total
				1	20.8		21.1		41.9
				8	17.0		22.6		**39.6**
				64	42.9		29.6		72.5
Multi-threading directives	MTA	2%	7%		Threads per processor				
				P	1	40	60	80	100
				1	150	3.82	2.72	2.22	2.04
				2		1.98	1.40	1.15	1.06
				4		1.01	0.74	0.64	0.59
				8		0.55	0.41	0.37	**0.35**

Fig. 2. Performance of the dynamic unstructured mesh-adaptation algorithm by platform and programming paradigm. All times are in seconds; the fastest times are highlighted.

Overset Grid-Generation Software Package

William Chan, Stuart Rogers

In computer simulations of flows about an object, a computational grid is used to model the object's geometry. The Chimera overset-grid method is currently one of the most computationally cost-effective options for obtaining accurate simulations of flow involving complex geometry configurations, viscous fluid dynamics, and bodies in relative, dynamic motion. Considerable success has been achieved in applying this method to a wide variety of problems. The objective of the current work has been to develop a comprehensive set of software tools for performing pre- and post-processing of overset grids for complex-geometry simulation problems, for both static and dynamic cases. These tools have been packaged together in the Chimera Grid Tools software.

The Chimera Grid Tools package allows a user to create overset computational grids, and to perform geometry processing, grid diagnostics, solution analysis, and flow-solver input preparation. This package has been requested by and distributed to over 200 U.S. organizations under nondisclosure agreements, and has been utilized in aerospace, marine, automotive, environmental, and sports applications. The software consists of a hierarchy of modules together with documentation and examples. At the lowest level are libraries containing commonly used functions such as input/output routines for data files, stretching functions, projection routines, and many others. At one level above the libraries are about 30 independent programs that can be used in batch mode. Capabilities offered by these programs include editing, redistribution, smoothing and projection of grids, hyperbolic and algebraic surface and volume grid generation, and Cartesian grid generation. At the highest level is a graphical user interface called OVERGRID.

The primary function of OVERGRID is to provide a single graphical interface environment for performing a wide range of operations prior to running the flow solver. These operations typically include checking and processing the input geometry, generating the surface and volume grids, analyzing the grid quality using various diagnostics, and creating input parameter files for the domain connectivity program and the flow solver. OVERGRID is used to create structured grids from a geometry description composed of multiple panel networks or triangles. Many of these operations are performed by the various individual modules of Chimera Grid Tools, and OVERGRID serves as a focal point from which these modules can be accessed.

The most recent addition to OVERGRID assists the user in preparing inputs for the OVERFLOW and OVERFLOW-D flow solvers. For dynamic simulations involving bodies in relative motion, the OVERFLOW-D input file may require up to 100 global parameters and about 70 to 90 parameters for each grid, depending on the numerical scheme chosen. Preparing such an input file for a complex configuration can be a formidable task, especially for a new user. The interface offered in OVERGRID provides a simplified set of commonly used and self-explanatory options for the user to choose from. Many parameters, such as topological boundary conditions, are automatically selected. Numerous error checks are also built in to ensure that a consistent and valid input file is created. An example using the Space Shuttle Launch Vehicle is shown in figure 1, which illustrates the kind of simulations that can be prepared

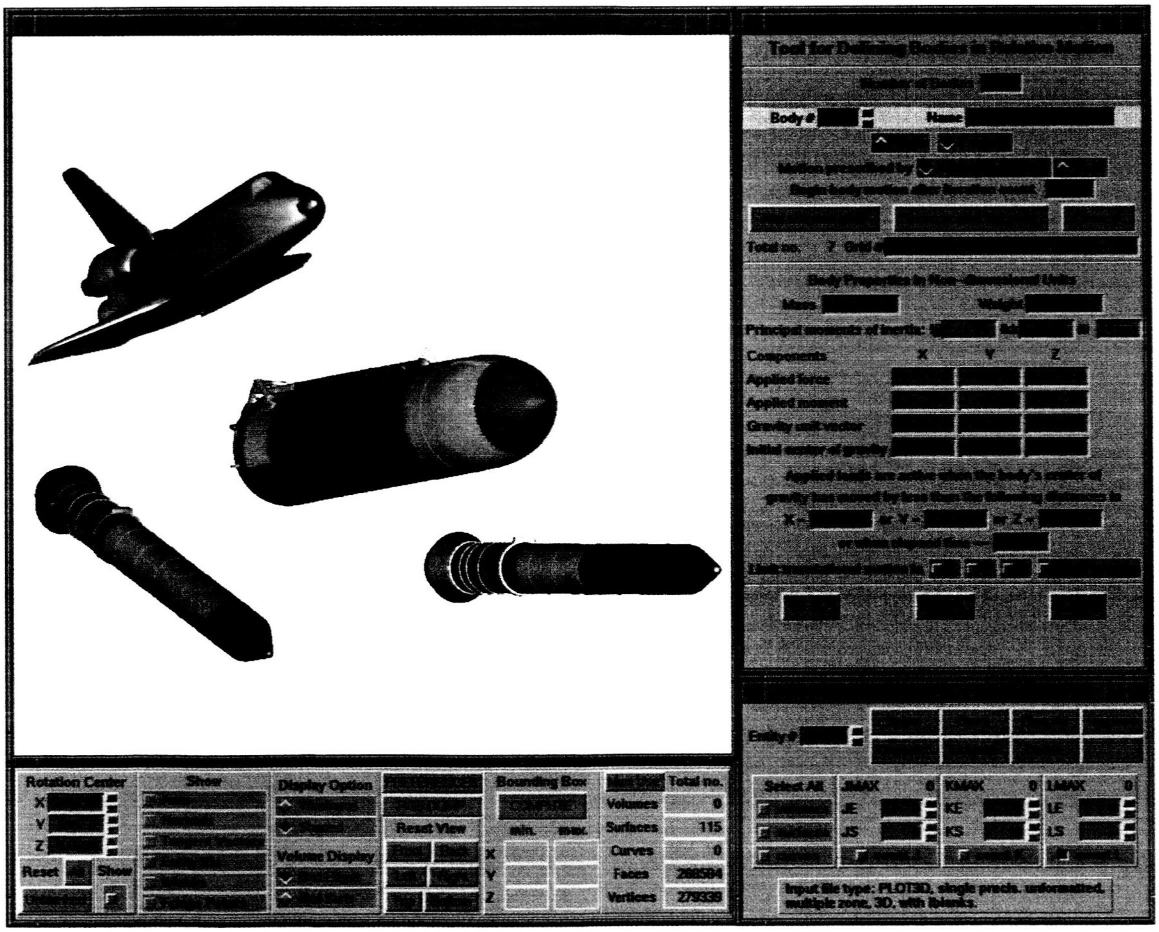

Fig. 1. An example Space Shuttle Launch Vehicle separation procedure.

using the new OVERFLOW-D interface in OVERGRID. Simple widgets allow the user to define the different bodies and to specify their attributes such as mass, moments of inertia, center of gravity, and applied loads.

Point of Contact: William Chan
(650) 604-6607
wchan@nas.nasa.gov

Multi-Level Parallel Computations of Unsteady Turbopump Flows

Cetin Kiris, Dochan Kwak

The objective of this effort was to provide a computational framework for the design and analysis of the entire fuel supply system of a liquid rocket engine, including high-fidelity unsteady turbopump flow analysis. This capability is needed to support the design of pump subsystems for advanced space-transportation vehicles that are likely to involve liquid propulsion systems. To date, computational tools for the design and analysis of turbopump flows are based on relatively lower fidelity methods. An unsteady, three-dimensional viscous flow analysis tool involving stationary and rotational components for the entire turbopump assembly has not been available for real-world engineering applications. The present effort provides developers with information such as transient flow phenomena at start up, the effect of nonuniform inflows, system vibration, and the effect on the structure.

In order to compute the flow on grids with moving boundaries, the overset-grid scheme was incorporated with the flow solver such that new connectivity data are obtained at each time step. The overlapped-grid scheme allows subdomains to move relative to each other, and provides a general flexibility when the boundary movement creates large displacements. Figure 1 shows the model for boost pump and the steps taken in the simulation procedure.

A parallel version of incompressible Navier-Stokes solver (INS3D) was developed by using

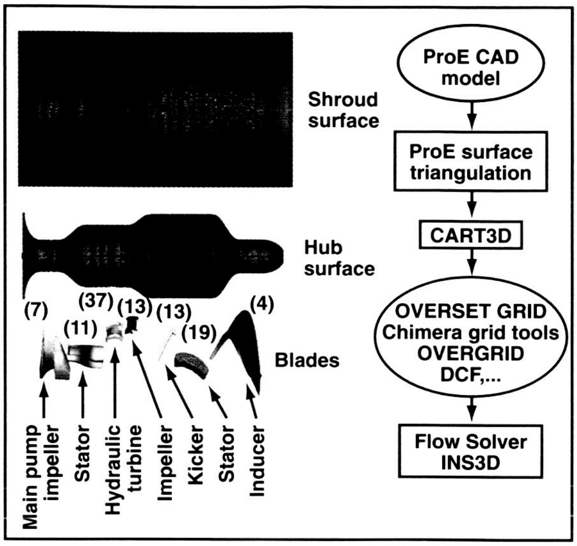

Fig. 1. Boost pump model and steps taken in the simulation procedure.

a Multi-Level Parallel (MLP) approach. This approach differs from the Message Passing Interface (MPI) approach in a fundamental way in that it does not use messaging at all. The coarsest level parallelism is supplied by spawning of independent processes via the standard UNIX fork. The boundary data for the overset-grid system is archived in the shared memory arena by each process. Other processes access the data from the arena as needed. Figure 2 shows INS3D-MLP performance versus computer processing unit (CPU) count for the 19.2 million-grid-point Space Shuttle Main Engine (SSME) impeller model.